

System Architecture Modeling for Technology Portfolio Management using ATLAS

Robert W. Thompson
Georgia Institute of Technology
School of Aerospace Engineering
270 Ferst Drive
Atlanta, Ga 30332
404-894-7783
robert_thompson@ae.gatech.edu

Daniel A. O'Neil
Marshall Space Flight Center
Huntsville, Alabama
Daniel.A.Oneil@msfc.nasa.gov

Abstract— Strategic planners and technology portfolio managers have traditionally relied on consensus-based tools, such as Analytical Hierarchy Process (AHP) and Quality Function Deployment (QFD) in planning the funding of technology development. While useful to a certain extent, these tools are limited in their ability to fully quantify the impact of a technology choice on system mass, system reliability, project schedule, and lifecycle cost. The Advanced Technology Lifecycle Analysis System (ATLAS) aims to provide strategic planners a decision support tool for analyzing technology selections within a Space Exploration Architecture (SEA). Using ATLAS, strategic planners can select physics-based system models from a library, configure the systems with technologies and performance parameters, and plan the deployment of a SEA. Key parameters for current and future technologies have been collected from subject-matter experts and other documented sources in the Technology Tool Box (TTB). ATLAS can be used to compare the technical feasibility and economic viability of a set of technology choices for one SEA, and compare it against another set of technology choices or another SEA. System architecture modeling in ATLAS is a multi-step process. First, the modeler defines the system level requirements. Second, the modeler identifies technologies of interest whose impact on an SEA. Third, the system modeling team creates models of architecture elements (e.g. launch vehicles, in-space transfer vehicles, crew vehicles) if they are not already in the model library. Finally, the architecture modeler develops a script for the ATLAS tool to run, and the results for comparison are generated.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. ATLAS OVERVIEW	2
3. CASE STUDY OVERVIEW.....	3
4. SYSTEM MODEL LIBRARY	5

5. EXAMPLE ARCHITECTURES	7
6. FUTURE CAPABILITIES.....	8
7. CONCLUSIONS.....	8
REFERENCES	8
BIOGRAPHY	9

1. INTRODUCTION

In response to the February 2004 "Vision for Space Exploration" and its stated goals of implementing a sustained and affordable human and robotic exploration program, the NASA Exploration Systems Mission Directorate established the Human and Robotics Technologies (H&RT) Programs. One tool developed under H&RT to analyze the impact of technology choices on an architecture is the Advanced Technology Lifecycle Analysis System (ATLAS). The "ATLAS Approach" to life-cycle system analysis is shown in Figure 1.

ATLAS can be used at several different levels. It can be used to study the mass and cost impact of a technology selection on a certain in-space vehicle or launch vehicle mass and cost during conceptual design. Using the Case Study framework, a series of System Models can be linked together to form a "campaign", or system level architecture. ATLAS may then be used as a decision support tool to analyze the mass and cost impact of technology choices on the architecture as a whole. Multiple architecture options can also be compared by building separate Case Studies.

Creating a model of a System Architecture involves several steps. The architecture requirements must first be defined, including identification of System Models required. Next the the technology options to be investigated are identified, after which a script for the execution of the Architecture within the ATLAS Framework is created, within which a

"ATLAS" Approach Advanced Technology Life-cycle Analysis System

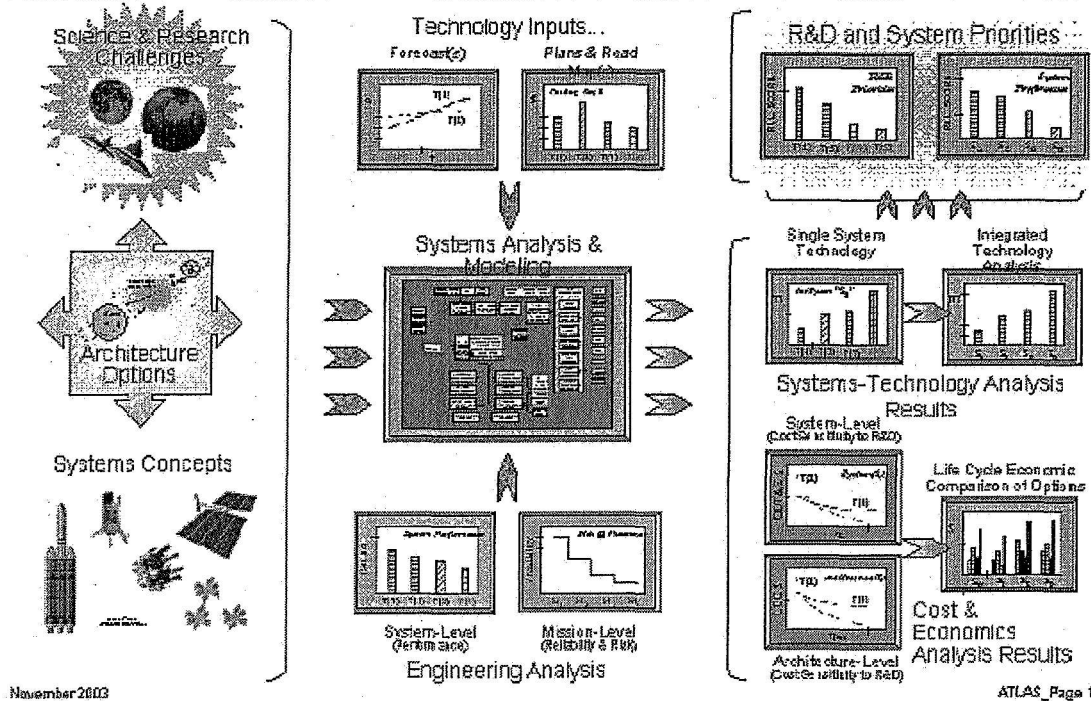


Figure 1 – ATLAS Overview [1].

campaign schedule (based on flight rate) for the lifecycle cost of the architecture is outlined.

2. ATLAS OVERVIEW

ATLAS has several elements that each provides an element of functionality for the overall tool. Because it is an Excel based tool, each element is a collection of Excel worksheets that interacts with the other Excel sheets via Visual Basic for Applications (VBA) code. Four Excel workbooks contain the worksheets and macros for the user interface: the controller, the integrator, and the accumulator. The Technology Tool Box (TTB) is a repository for current and future technology performance data that is used to analyze different technology funding scenarios. The Model Library is the collection of system models that make up the options for vehicles and surface elements that can be modeled for cost and mass with different technologies. Finally, each Case Study provides a script for ATLAS to run a collection of System Models in a certain order, simulating a certain architecture for accomplishing a goal (e.g. putting three men on the moon) once or multiple times.

ATLAS Framework

ATLAS is made up of many Excel workbooks, but for a beginner there are a few important worksheets that one

should be familiar with. One important task of the ATLAS Framework is to coordinate the transfer of data between the various ATLAS elements.

The ATLAS User Interface (AUI) is the first thing the user sees upon opening ATLAS. It provides a starting point from which to load case studies, analyze individual vehicles, and view the mass and cost results of the analyses. Individual System Models may be run through the ATLAS User Interface (AUI) via a pull-down menu on the ATLAS toolbar (Figure 2). The AUI loads a default Case Study whenever it is opened. The user can load a new Case Study, save the current Case Study, or run a Script from the ATLAS toolbar (Figure 2).

The code for the data transfer is mostly contained in the SIAM_Controller worksheet. The basic functionality for the SIAM_Controller is as follows:

1. Open the System Model (selected either in Menu Mode by the user or as part of a Script)
2. Transfer user input data into the System Model
3. Transfer technology data into the System Model from the TTB
4. Run code within the System Models (if any).
5. Transfer model outputs (system mass breakdown and other information needed for cost estimation) to other worksheets for cost estimation

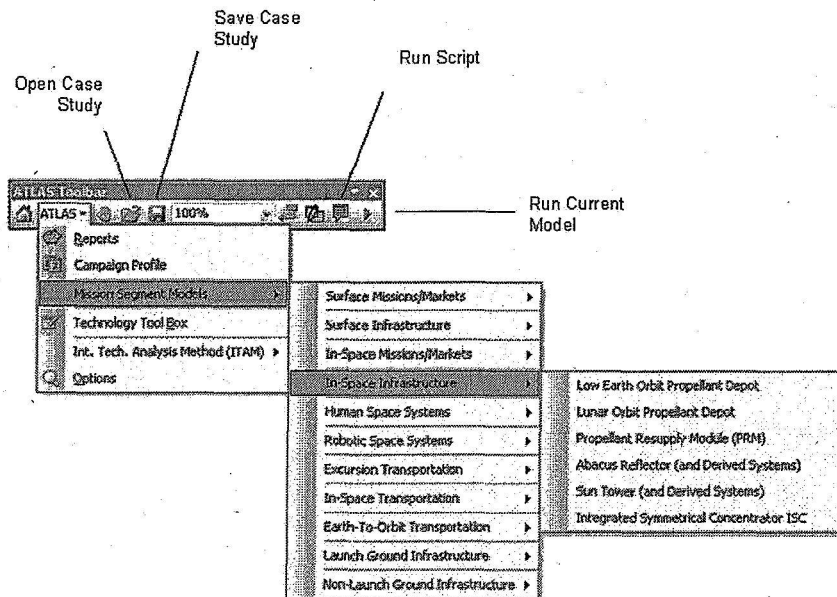


Figure 2 - ATLAS Toolbar.

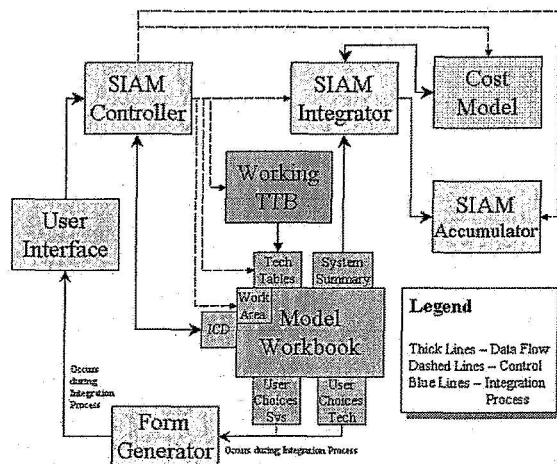


Figure 3 - Model Execution Flow Diagram [1]

6. Close the System Model (without saving), and open the next model for analysis.

A more detailed flow diagram of this process is shown in Figure 3.

Technology Tool Box

The Technology Tool Box is of importance to System Model and Case Study developers because it is the repository for technology metric information within ATLAS. The TTB is organized according to a WBS developed by John Mankins [REF] to categorize technologies in a logical

and functional manner. Each technology has a series of metrics that define its performance capabilities. For example, a rocket propulsion technology will have a Thrust metric, and a Specific Impulse metric, among others. Technology forecasting by technologists and disciplinary specialists has also been performed on some technologies and metrics, and this data is made available through the use of the Time Frame concept. The ATLAS "Time Frame" goes from 0 to 10, with 0 being today (2005). Each step represents three years. Therefore, Time Frame 10 contains the state of the art technology metrics for 2032. In theory, if proper forecasting on a metric has been performed, a metric should improve in performance if funding is being applied, or stay the same if there is no funding for that particular technology.

In order to analyze the impact of a particular technology choice on architecture, the System Architect should first check the TTB to see if it has been included and all necessary data is present. The TTB is still a work in progress, and values for all metrics have not yet been entered.

3. CASE STUDY OVERVIEW

Each Case Study is a set of instructions for the ATLAS Framework to run a simulation of architecture. The most important parts of the Case Study are the introductory cartoon of the architecture, the Interface Control Data, the Script, the Mission Definition, and the Campaign Profile.

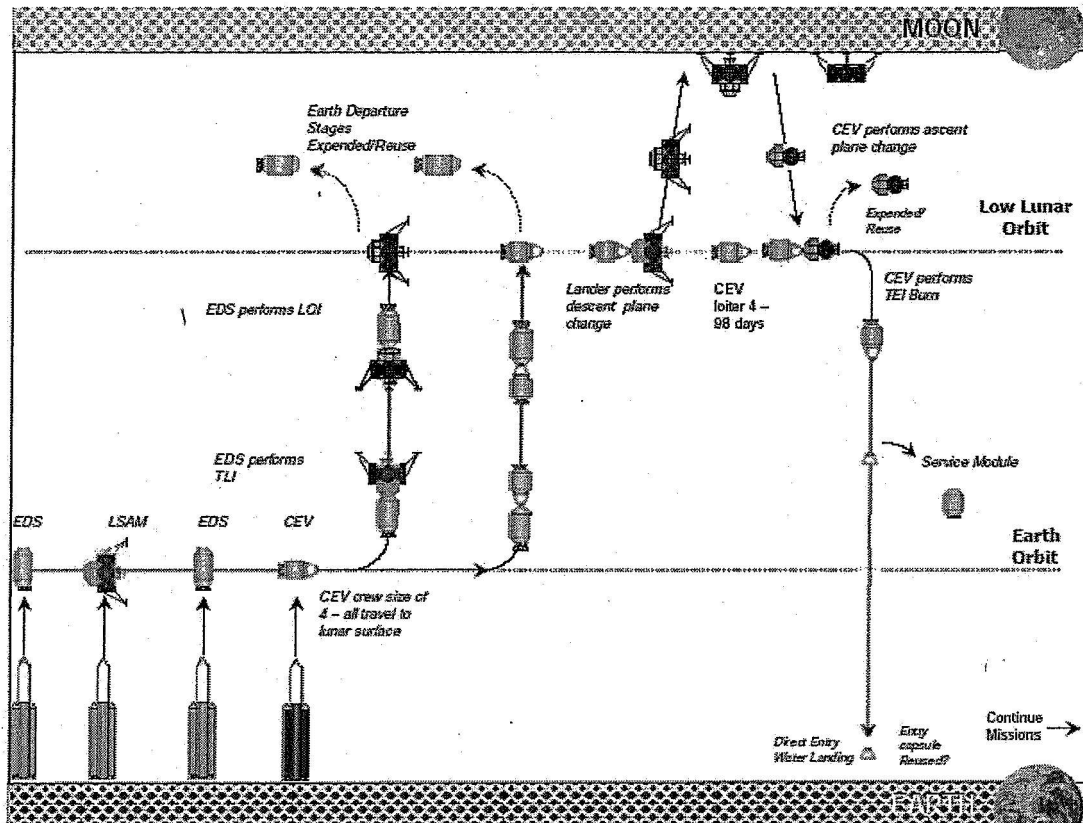


Figure 4 - Point of Departure Architecture [2]

Introduction

An important part of the Case Study is the introductory cartoon of the architecture that is being modeled. A sample cartoon is shown in Figure 4. This format is preferable for portraying Moon and Mars exploration missions because of the ability to show all of the required launches, as well as all of the necessary in-space elements required by the architecture.

Interface Control Data

The Interface Control Data (or ICD) sheet of each Case Study should show each input and its default value for each model used in the Case Study. The purpose of this sheet is to provide a link between the Case Study and the AUI for the transfer of input data in the Script. When the user loads a Case Study through the AUI, the Case Study ICD is copied into the AUI, from which inputs are copied by the SIAM Controller during the execution of a Script. The ICD also serves as a quick reference for the Case Study author to verify input names and values when making the Script page.

Script

Once an architecture has been defined, it is broken into "missions" (see *Mission Definition* below). This is usually

done by making each mission a single launch from Earth. The models that are run for each mission include the payloads and the launch vehicle. Because the launch vehicle sizing depends on the size of its payloads, the models are typically run in "top-down" order. For example, a single Apollo mission could be accomplished with a single mission, because there was only one Saturn V launch per lunar mission. The models would therefore be run in the following order:

1. Lunar Module
2. Command Module
3. Service Module
4. Saturn V

In this case, it is important that the Command Module be sized before the Service Module, because the propellant mass in the Service module depends on the mass of the Command Module.

The Script provides a place to link together these dependent payload masses so that the outputs of one or more models

A	B	C	D	E	F	G	H	I	J
Total Wet Dry	Manifest 1 440 440	Manifest 2 20474.0 6628.0	Manifest 3 5587.0 5089.0	Manifest 4 13892.0 4574.0	Manifest 5 71569 7134	Manifest 6 51864.0 5706.0	Manifest 7	Manifest 8	Manifest 9
Mission 3	Label	Manifest #	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7
Eagle Centurion	LSAM Centurion	2 10	440 19910.4	100 N	14 8	0 30	0 RD180	1.5	4
Mission 4									
Capsule PPM Delta4	Capsule PPM Delta4	3 4 11	Reentry Vehicle 3 LED	4 3 100	14 5 13892.0	9 5387.0 5	0 7	4 4	Solar Arrays Y
Mission 1									
Manticore Centurion	EDS1 Centurion	5 12	3 51658.6	3 N	1 8	19910.4 30	0 RD180	Y	Y
Mission 2									
Manticore Centurion	EDS2 Centurion	6 13	3 37972.0	3 N	1 8	13892.0 30	0 RD180	Y	Y
End Script									
		time on LS/route							
		crew number							
		CEV Volume							

Figure 5 - Sample Case Study Script

[illegible]

Figure 6 - Sample Mission Definition

(Command and Service Module Gross Mass) become the inputs to the next model (Saturn V Payload Mass). This is achieved through the "Manifest" fields at the top of the Script page. The Script is also where common inputs between models can be entered, maintaining consistency in the architecture. For example, the Command Module model may require an input for "Number of Crew" in order to size the required volume of the crew capsule; likewise, the Service Module may require the same input in order to size life support or power systems. Consistency between these models is ensured using a single page for such inputs shared between models.

The Script page is also the place to change technology choices. All model inputs are shown on the Script page, and can be confirmed with the ICD page or with the model itself (the Script page displays only the current value of each input, but the inputs are numbered, rather than named – hence the need for the ICD page, which shows the input names and values for all models).

If, perhaps, the author of a Case Study enters a value into the Case Study Script that is either 1) out of range of that

particular input, or 2) not a valid option as specified by the model, the Script will run with the default value for that input, and an error will be generated and reported to the user after the Script has run. If, for example, the author or analyst makes a Script that analyzes a 5-person Apollo Architecture, if the launch vehicle payload (Lunar Module and Command and Service Module mass) exceeds the maximum payload of the Saturn V, an error will be reported to the user.

The Case Study Script, shown in Figure 5, is organized into missions corresponding to the Point of Departure architecture shown in Figure 4.

Mission Definition and Campaign Profile

The Mission Definition page is useful for laying out an architecture. A sample mission definition is shown in Figure 6. If the Case Study author wishes to see the a life cycle cost curve, he or she writes a campaign profile that establishes the launch dates for each separate mission in the campaign. If several flight rates are being considered (one mission per year versus four missions per year), multiple campaign profiles can be generated and switched between via the

"Crews Per Year" input on the "Missions" page of the Case Study workbook.

4. SYSTEM MODEL LIBRARY

The ability of ATLAS to accurately model a Space Exploration Architecture depends on the availability of ATLAS models that are representative of the elements of that SEA. The System Model Library is the collection of physics-based and historical-based models that have been produced for ATLAS. These models fall into several categories:

1. Historical Vehicles (examples: Space Shuttle, Eagle Lunar Lander, Saturn V)
2. Conceptual Launch Vehicles
3. Conceptual In-Space Transportation Vehicles
4. Conceptual Crew Capsules and Entry Vehicles
5. Conceptual Surface and In-Space Infrastructure

Models have been developed by engineers at Marshall Space Flight Center, Johnson Space Center, the Jet Propulsion Laboratory, SAIC, and the Georgia Institute of Technology.

Historical Vehicles

Among the historical vehicles modeled in ATLAS are several launch vehicles, including the Space Shuttle, Delta IV, Atlas V, and the Saturn V (for comparison with the Apollo architecture). These historical vehicles are important for calibration of ATLAS cost estimation methodology used for other conceptual models. For example, if the Apollo architecture is modeled, the estimated cost should match the real cost of the Apollo program. In some cases, existing launch vehicles have also been included in future architectures, such as the potential use of EELV (Delta IV and Atlas V) for use in potential lunar architectures.

Other historical vehicles that have been included in the ATLAS model library are the Lunar Module, which can be used to compare against new two-stage expendable lunar lander models (such as the Lunar Surface Access Module – LSAM), and the Command and Service Module (CSM) from the Apollo program.

Conceptual Launch Vehicles

Several conceptual launch vehicles have been designed, principally by the Georgia Tech model development team. These designs are typically parametric models of single point designs efforts performed by the Space Systems Design Lab (SSDL) at Georgia Tech. These models generally utilize propulsion and structural technology

information from the TTB, allowing the user to vary those parameters and see the resulting impact on performance. The Centurion conceptual expendable launch vehicle is shown in Figure 7. It is a cargo launch vehicle that utilizes shuttle-derived elements (such as solid-rocket boosters) [3].

Conceptual In-Space Transportation Vehicles

The in-space transportation models that have been developed include chemical upper stages for lunar transfer, similar to the Saturn IVB and solar electric transfer vehicles for lunar or Mars transfer. Another model that has been developed is the Artemis reusable lunar lander, which has options for refueling in lunar orbit or on the lunar surface (depending on the infrastructure options available) [4]. The Artemis lander is shown in Figure 8.

Conceptual Crew Capsule and Entry Vehicles

These models have been developed to simulate a number of crew module configurations, including Apollo-style capsules, as well as next-generation biconic configurations. Key technologies include power generation, structures, avionics, and environmental control. Tempest is an example of a next-generation reusable CEV for lunar missions [5], and is shown in Figure 9.

Conceptual In-Space and Surface Infrastructure

The infrastructure models are chiefly power generation and propellant storage. Many architectures featuring reusable transportation systems require either pre-deployment of propellant, or in-situ generation of propellant. In either case, a storage facility is required, and frequently requires the capability to store cryogenic propellants and propellants for electric propulsion (such as Xenon or Krypton). Other infrastructure models include the Space Solar Power model, and there are plans for models of surface power and propellant generation facilities.

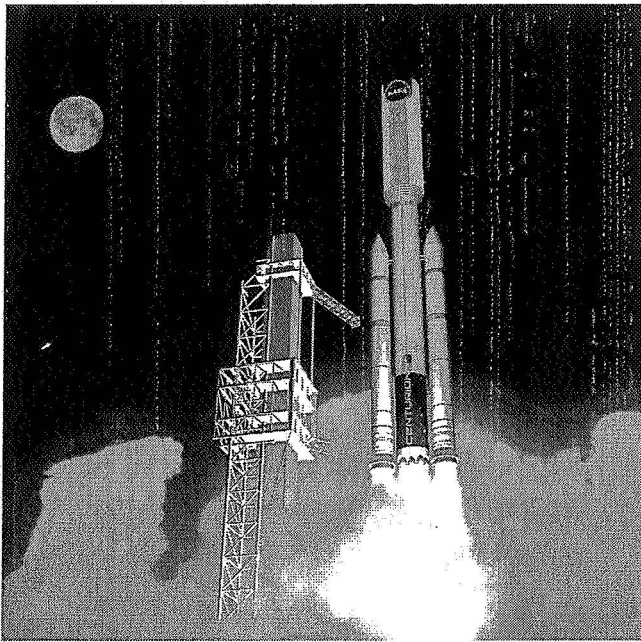


Figure 7 - Centurion Conceptual Launch Vehicle [3]

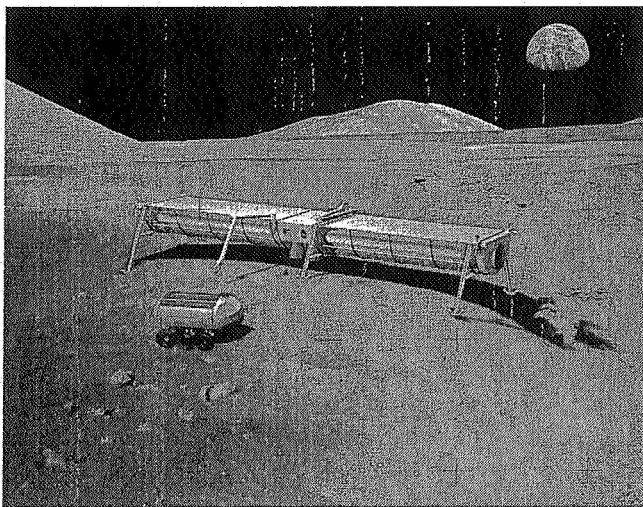


Figure 8 - Artemis Reusable Lunar Lander [4]

5. EXAMPLE ARCHITECTURE

Among the lunar exploration architectures that have been investigated using ATLAS, the Point of Departure architecture (produced by the requirements division of the Exploration Systems Mission Directorate at NASA Headquarters) is one the simplest, and is a logical architecture given the current technologies and launch vehicles currently available. The architecture, shown in Figure 4, utilizes four launches per manned lunar mission. These launch vehicles have been modeled as Centurion launches, because it is a conceptual non-human-rated launch vehicle that utilizes shuttle derived components (namely the

Shuttle SRBs). The crew-launch is baselined on a human-rated EELV (Delta IV). The results from ATLAS for the POD architecture using all LOX/LH₂ propulsion systems and current technologies for power and structures are shown in Figure 11.

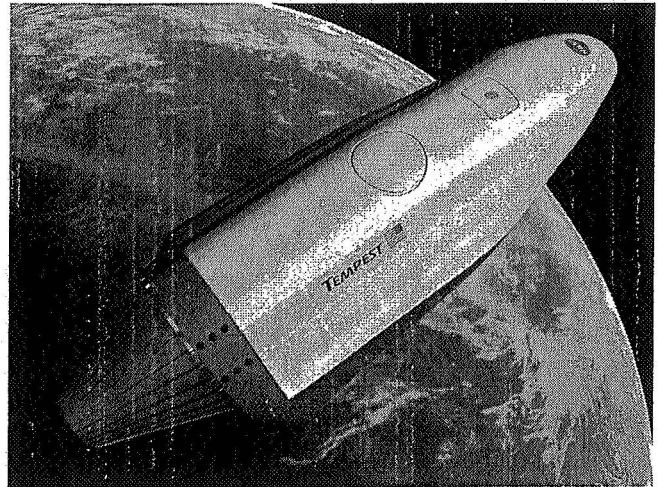


Figure 9 - Tempest Reusable Crew Exploration Vehicle [5]

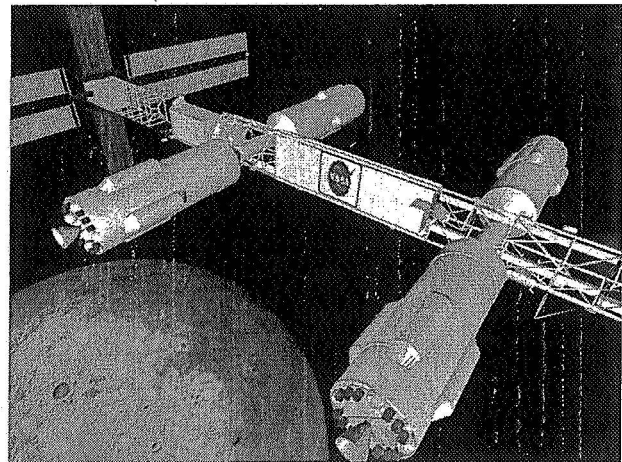


Figure 10 - Low Lunar Orbit Depot [6]

The first and second "humps" show the cost of development of the systems and the production costs respectively. The cost of each system as well as technology developments within the various system appear on this lifecycle cost chart.

In order to perform a trade study, an analyst could change technology or other system options (propulsion system, number of crew, etc.) in the Case Study script, and then rerun the ATLAS simulation.

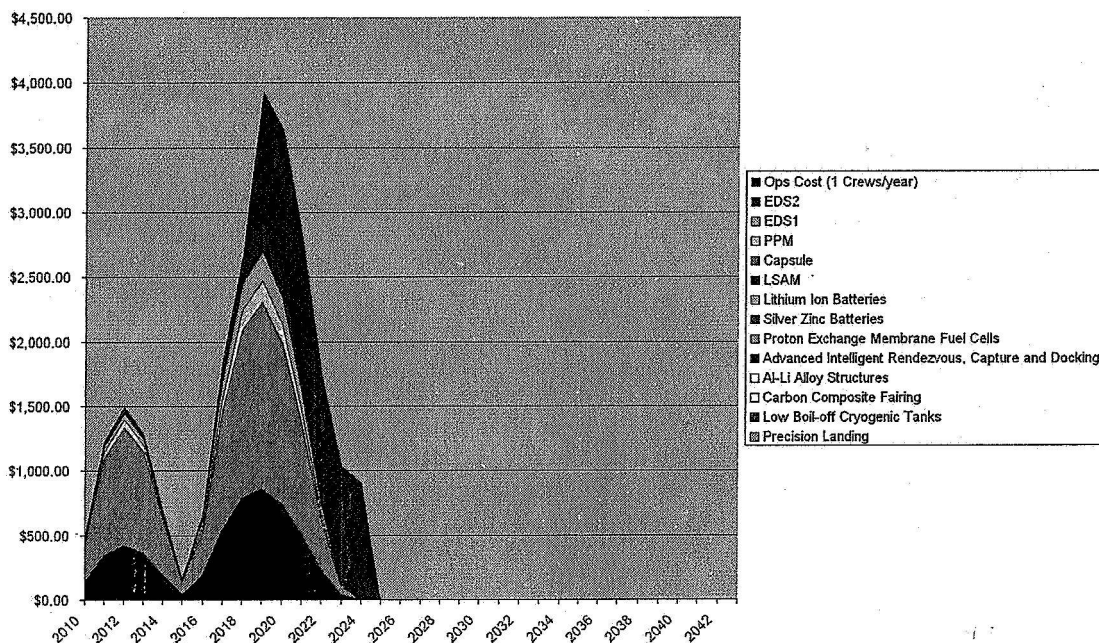


Figure 11 – POD Architecture Life-Cycle Cost

6. FUTURE CAPABILITIES

The current exclusively Excel based version ATLAS is a prototype. Certain elements of ATLAS lend themselves to different formats, which would make the execution of the tool more efficient. For example, the TTB may eventually be migrated to a database, rather than the 10+ megabyte Excel sheet currently in use. Additionally, the script functionality of the Case Study may be replaced with a discrete-event-simulation code written in the Java-based Ptolemy II environment (developed at UC Berkeley). The important aspect of these future environments is that they should be able to interact with other parts of ATLAS that are well suited to Excel, such as the system models, and ensuring this capability is a current focus of the ATLAS team. Another current task is the creation of a “what-you-see-is-what-you-get” style Case Study generator. This will allow the faster generation of new architectures, and simplify the process of performing trade studies on current architectures.

7. CONCLUSIONS

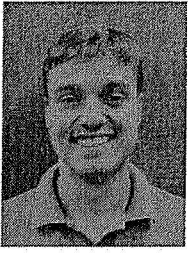
In making decisions for strategic planning, there is a need for physics and historical based tools that can make predictions of performance and cost quickly and inexpensively. In replacing consensus based methods that could be tainted by the bias of the individuals involved, ATLAS is limited only by the technologies in the TTB and

the models in its model library. Using ATLAS will ensure that all analyses utilize the same technology information and the same cost models, making direct comparison between analyses possible.

REFERENCES

- [1] Daniel O’Neil and John Mankins. The Advanced Lifecycle Analysis System (ATLAS). 2004 International Astronautical Conference. IAC-04-IAA.3.6.3.01.
- [2] Michael F. Lembeck. ESMD Requirements Division. Requirements Process Overview. February 2005.
- [3] David A. Young, John R. Olds, et al. Centurion: A Heavy-Lift Launch Vehicle Family for Cis-Lunar Exploration. AIAA 2004-3735.
- [4] David A. Young, John R. Olds, et al. Artemis: A Reusable Excursion Vehicle Concept for Lunar Exploration. AIAA 2005-4010.
- [5] Virgil Hutchinson, John R. Olds, et al. Tempest: Crew Exploration Vehicle Concept. AIAA 2005-4190.
- [6] Don Perkinson. Low Lunar Orbit Depot ATLAS Model.

BIOGRAPHY



Robert Thompson is a Research Assistant in the Space Systems Design Lab of the Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology. He has worked two summers at Marshall Space Flight Center on the ATLAS Project.



Daniel O'Neil manages the ATLAS project. In 1985, he graduated from the University of Alabama in Huntsville with a B.S. Degree in Electrical and Computer Engineering. In 1998, he received his M.S. Degree in Engineering Management. Since 1990, he has been employed by NASA to manage space exploration advanced

concept analysis activities.



Space Systems Design Lab
Georgia Tech Aerospace Eng.

System Architecture Modeling for Technology Portfolio Management using ATLAS

Robert Thompson
Space Systems Design Laboratory
Georgia Institute of Technology

Daniel O'Neil
Advanced Concepts Office
Marshall Space Flight Center

March 6, 2006



ATLAS Overview

- Advanced Technology and Lifecycle Analysis System
- ATLAS is a technology evaluation and costing tool for use by NASA headquarters to evaluate the impact of new technologies on NASA missions
- ATLAS is Excel based
 - Set to run on both a PC and a Mac



ATLAS Team

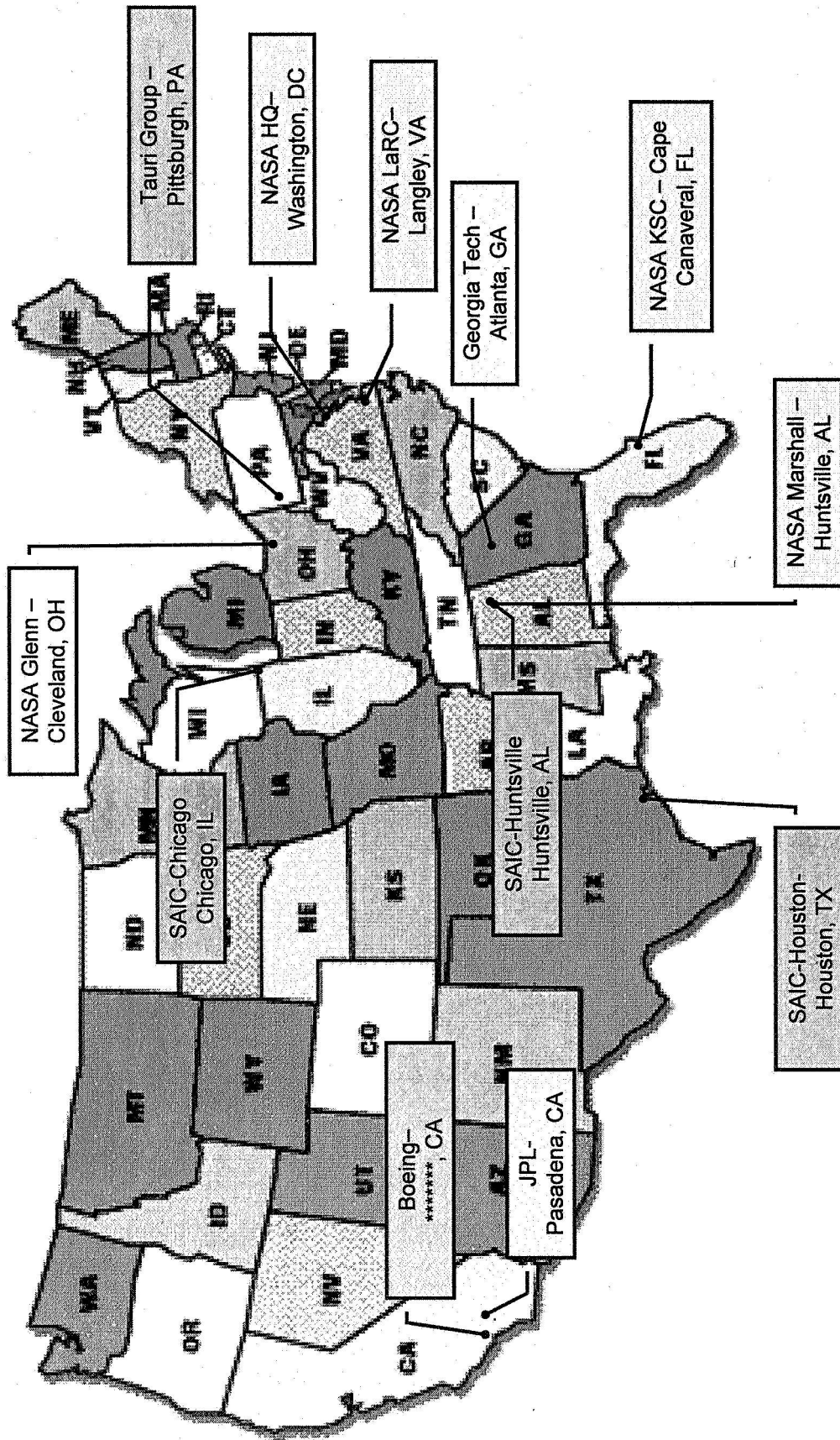
- ATLAS team headed by Dan O'Neil MSFC
- Head Architect - Harvey Feingold SAIC
- Model Developers
 - Georgia Tech
 - KSC
 - JPL
 - SAIC
 - Boeing
- Validation- Pat Troutman NASA LaRC



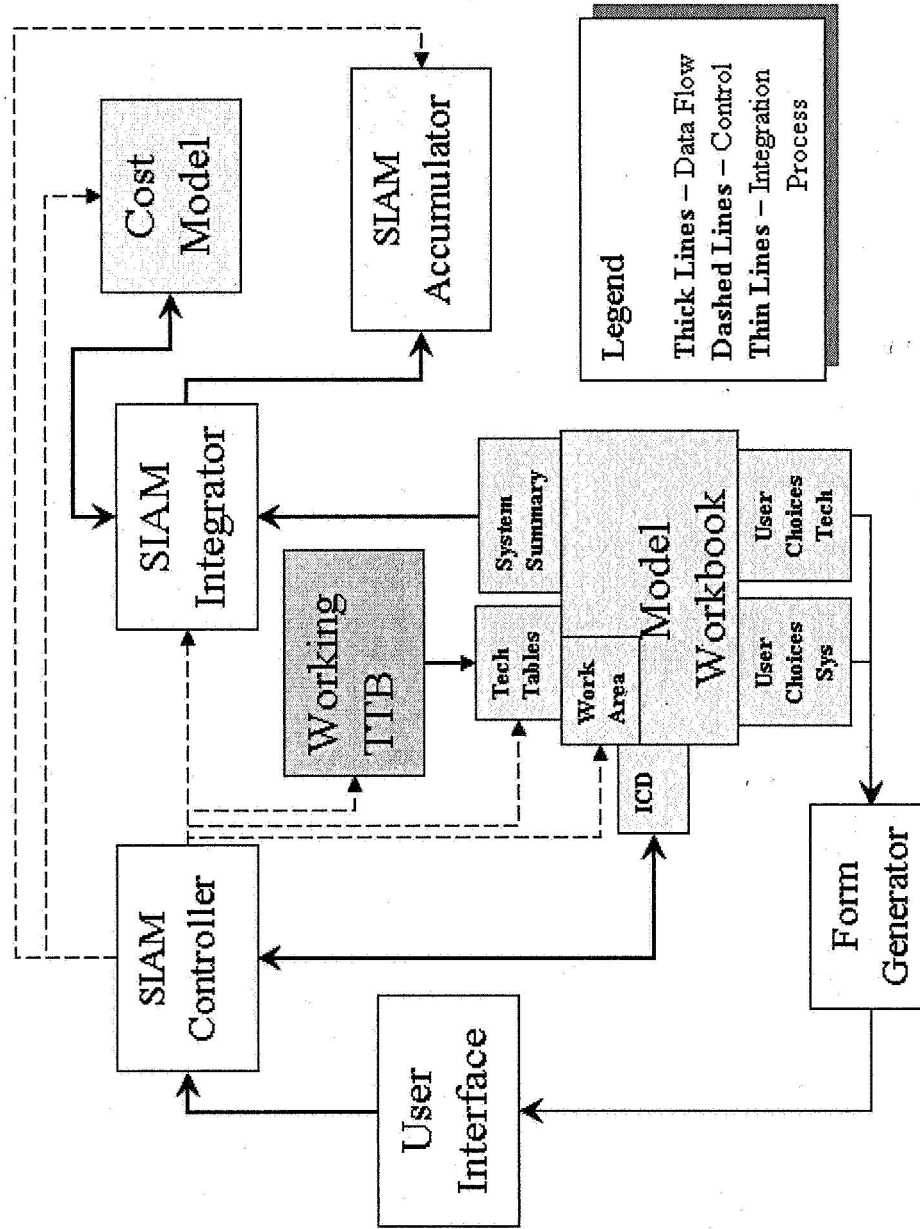


Space Systems Design Lab
Georgia Tech Aerospace Eng.

ATLAS Team



ATLAS Flow



ATLAS

- Choose campaign profile, technologies, and vehicles to be used
- ATLAS SIAM opens all of the vehicles and passes the necessary information (technologies, masses, etc...)
- Results:
 - Mass breakdown
 - Life Cycle Cost breakdown
- Can combine a variety of models for a wide range of architectures
 - Gives a quick comparison between different architectures for same mission requirements

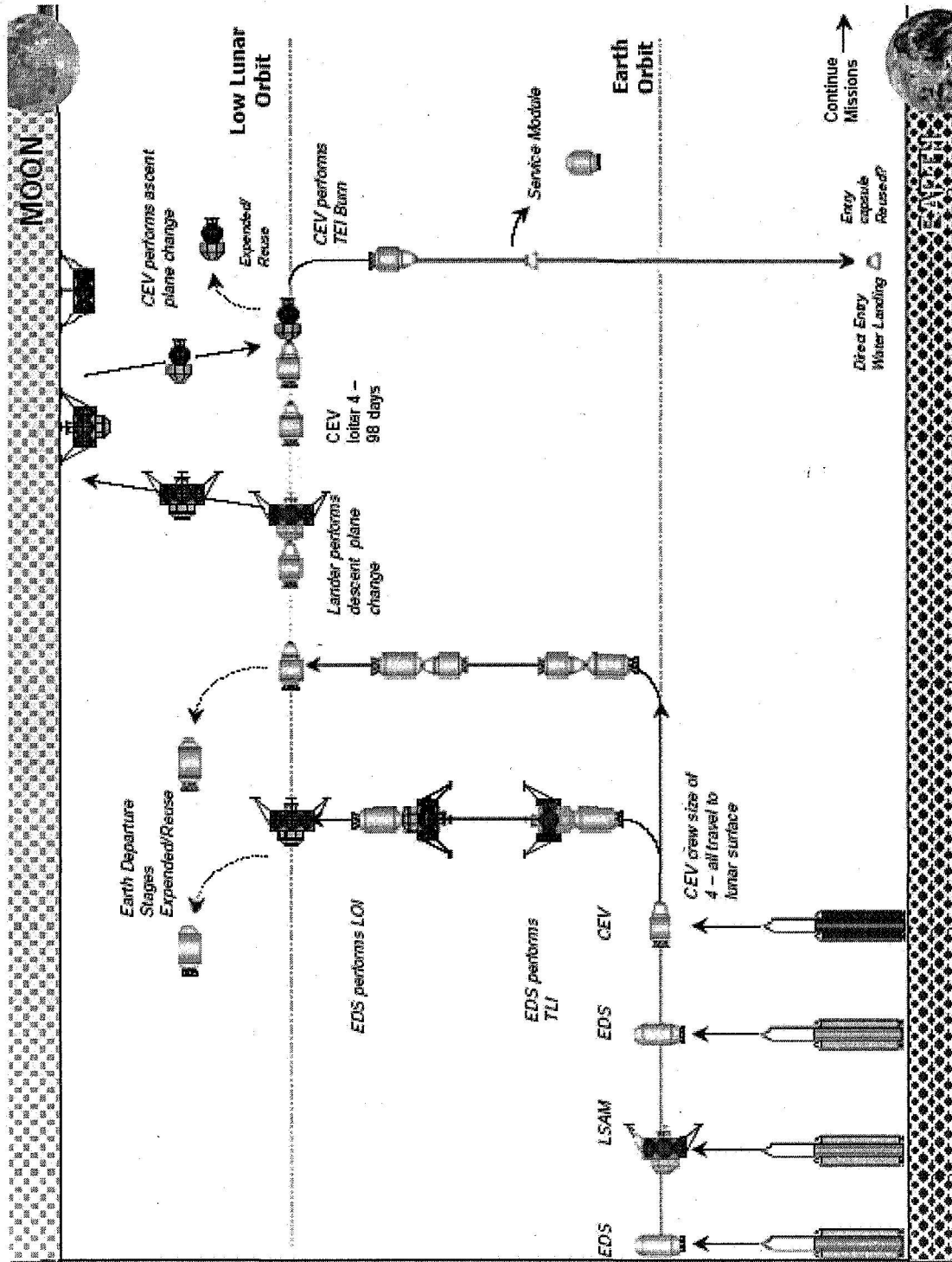


Project Tasks

1. Develop system architecture “Case Studies” that allow the sensitivity of technology choices to be analyzed
 - Models are sensitive to:
 - Technology changes
 - Changing payloads
 - Design choices (fuel type, number of arrays, etc.)
 - Model outputs are used to size other vehicles in scenario
2. Provide overall support to ATLAS model integration and testing



Sample Case Study: Point of Departure





Space Systems Design Lab
Georgia Tech Aerospace Eng.



Space Systems Design Lab
Georgia Tech Aerospace Eng.

Mission Definition

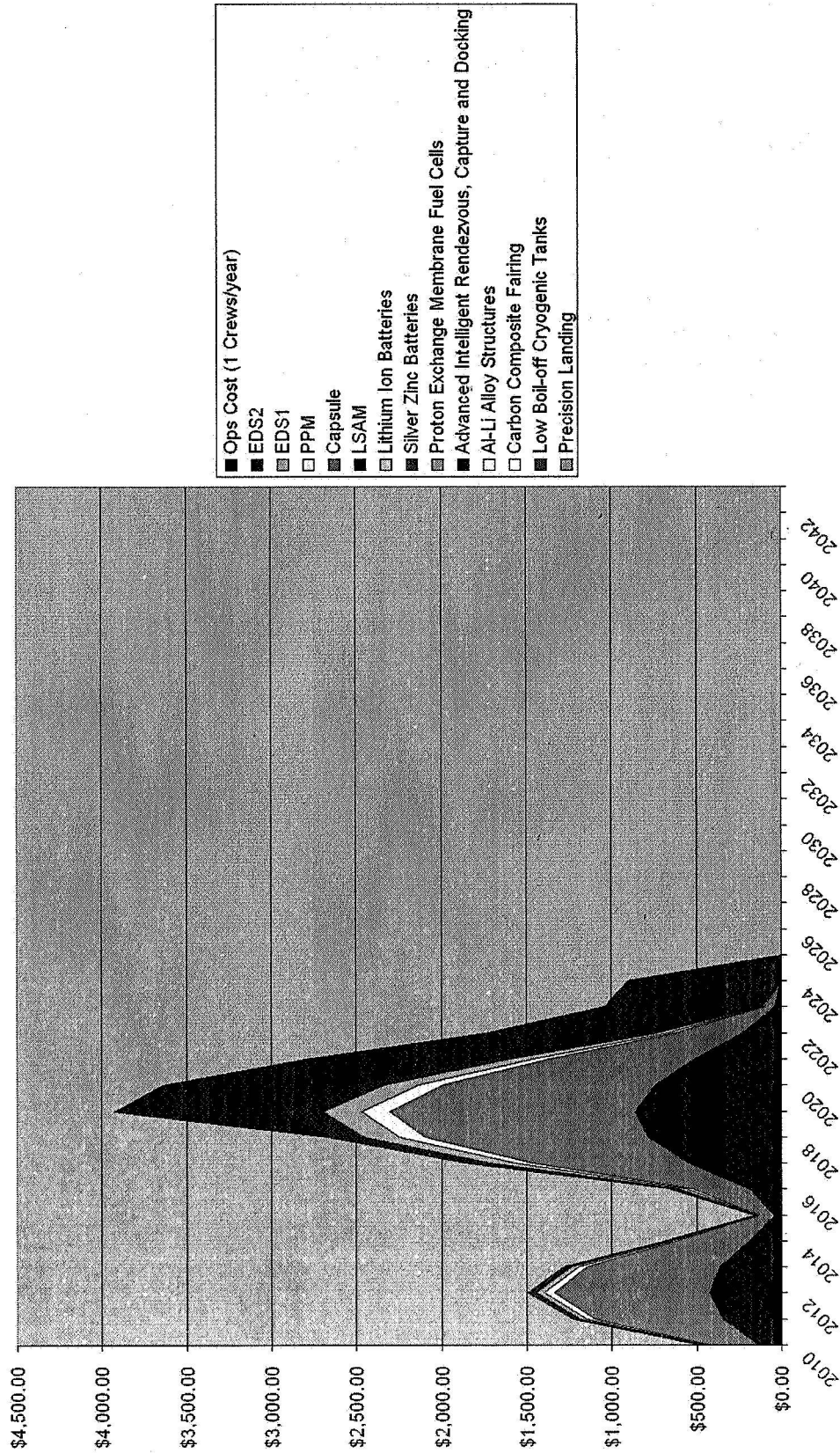
MISSION	4	Deliver Crew to Moon and Back	No. Launches	System	Payload 1	Payload 2	Location	Destination
Event No.	1	Launch	1	Delta4	Capsule	PPM	KSC	LEO
2	-	Orbit Insertion		Capsule	PPM		LEO	LEO
3	1	Rendezvous		EDS2	Capsule	PPM	LEO	LEO
4	3	Space Transfer		EDS2	Capsule	PPM	LEO	Moon
5	-	Orbit Insertion		EDS2	Capsule	PPM	Moon	LLO
6	1	Rendezvous		Capsule	LSAM	PPM	LLO	LLO
7	-	Crew Transfer		LSAM	Capsule	PPM	LLO	LLO
8	3	Deorbit/Land		LSAM			LLO	Lunar Surface
9	1	Launch		LSAM			Lunar Surface	LLO
10	1	Rendezvous		LSAM	Capsule	PPM	LLO	LLO
11	-	Crew Transfer		LSAM	Capsule	PPM	LLO	LLO
12	-	Separate		Capsule	PPM		LLO	LLO
13	3	Space Transfer		Capsule	PPM		LLO	Earth
14	-	Separate		Capsule			Earth	Earth
15	-	Enter/Land		Capsule			Earth	Earth Surface
16	-						Earth Surface	





Space Systems Design Lab
Georgia Tech Aerospace Eng.

Results: Life Cycle Cost



March 6, 2006



*Space Systems Design Lab
Georgia Tech Aerospace Eng.*

Model Library

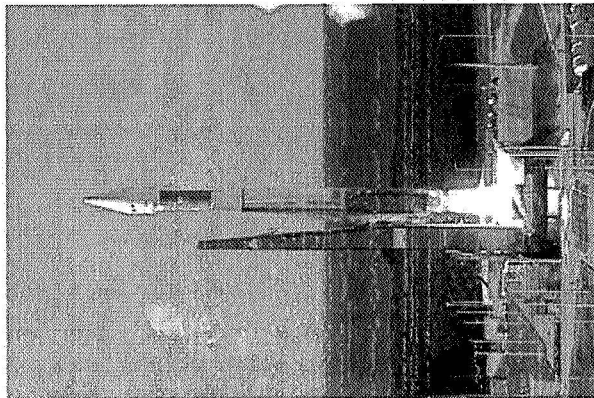
- Transportation Responsibilities
 - Excursion Vehicles Transportation Segment
 - In-Space Transportation Segment
 - Earth To Orbit Transportation Segment
 - Ground Launch Infrastructures



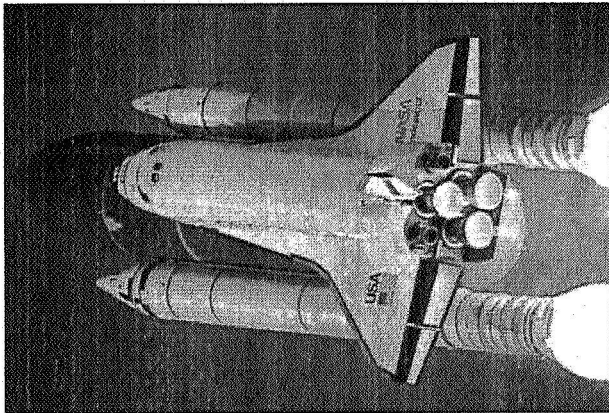


Space Systems Design Lab
Georgia Tech Aerospace Eng.

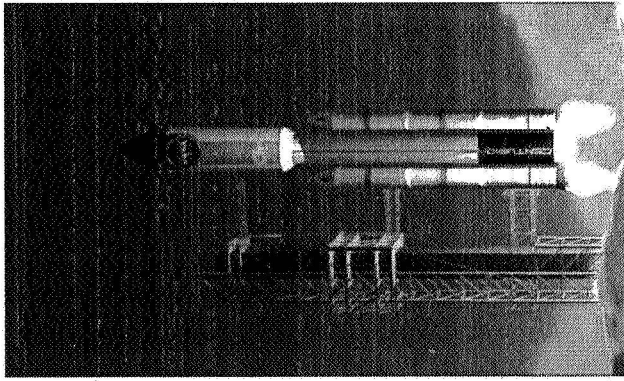
ETO Vehicles



EELVs

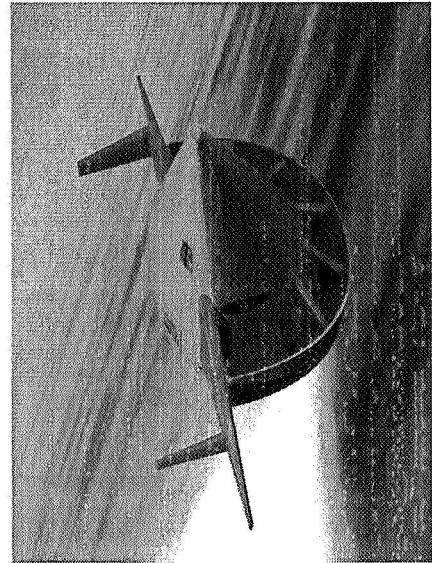


Space Shuttle

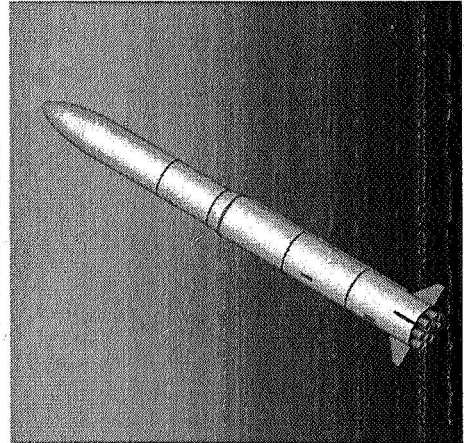


Centurion*

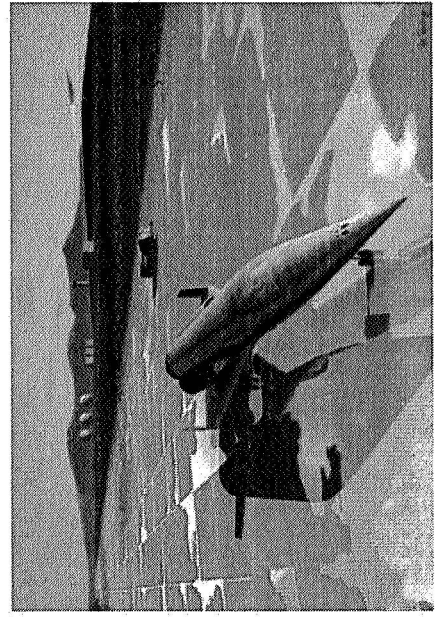
Hyperion



Colossus

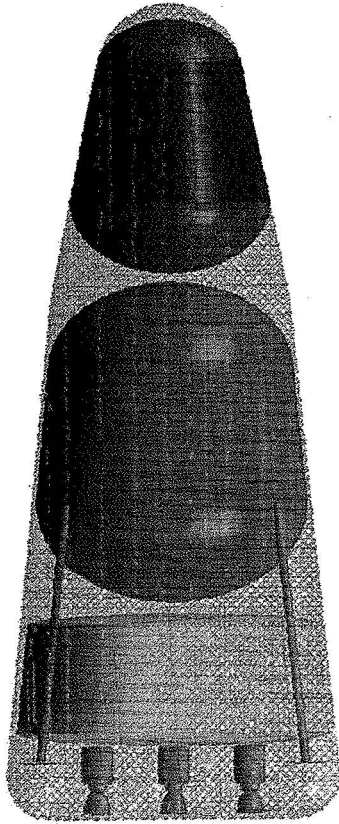


Argus*

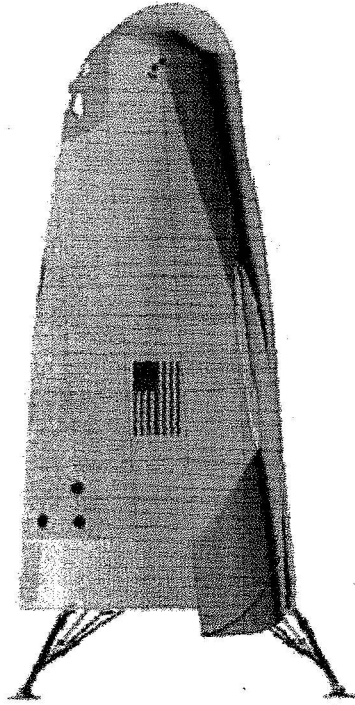


Excursion Vehicles

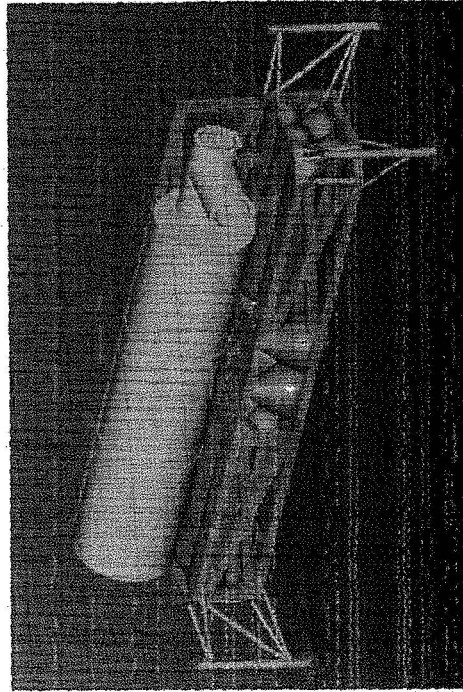
MREV



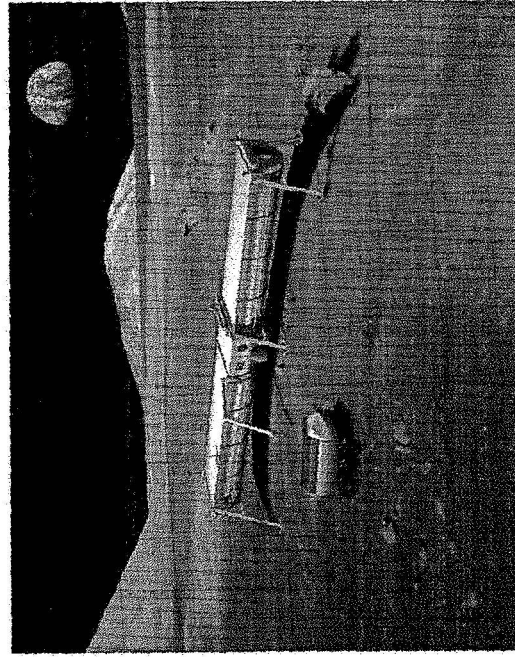
DSSS*



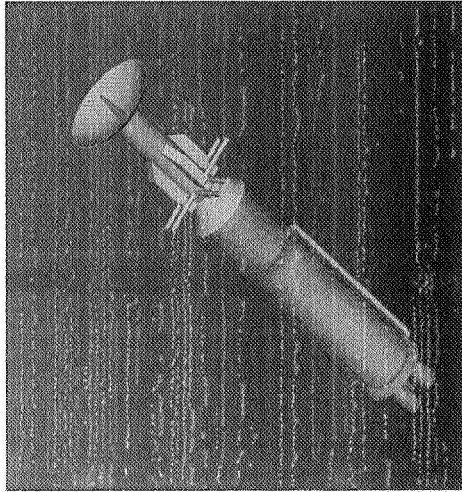
Reusable Lunar Lander



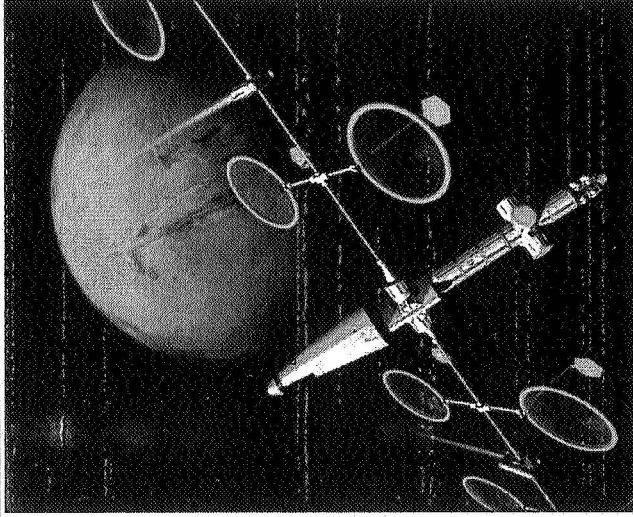
Artemis



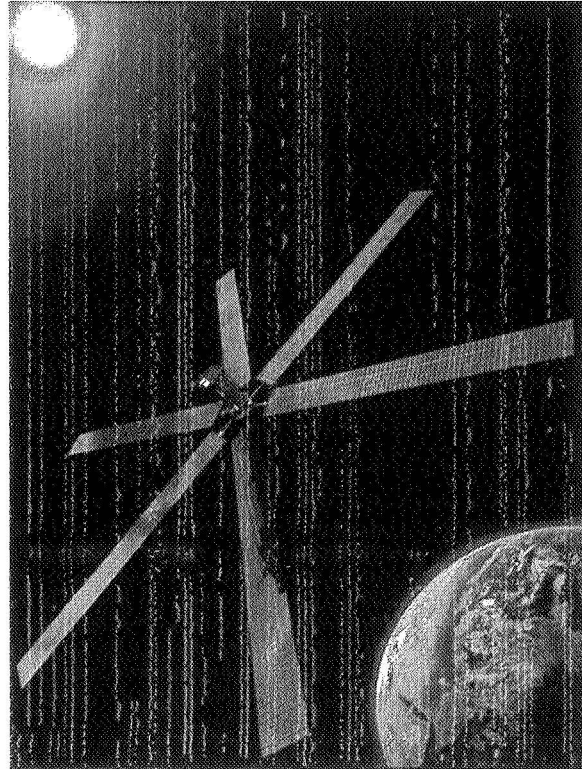
In-Space Vehicles



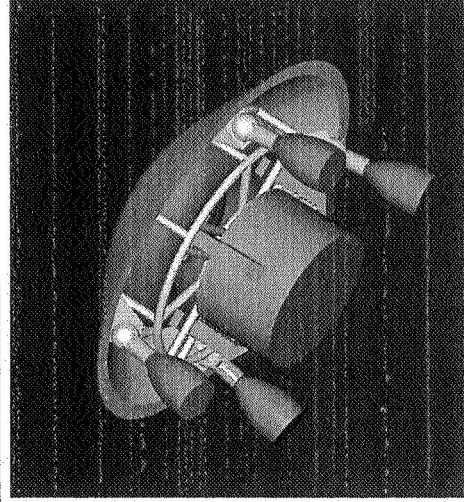
Manticore



Solar Clipper



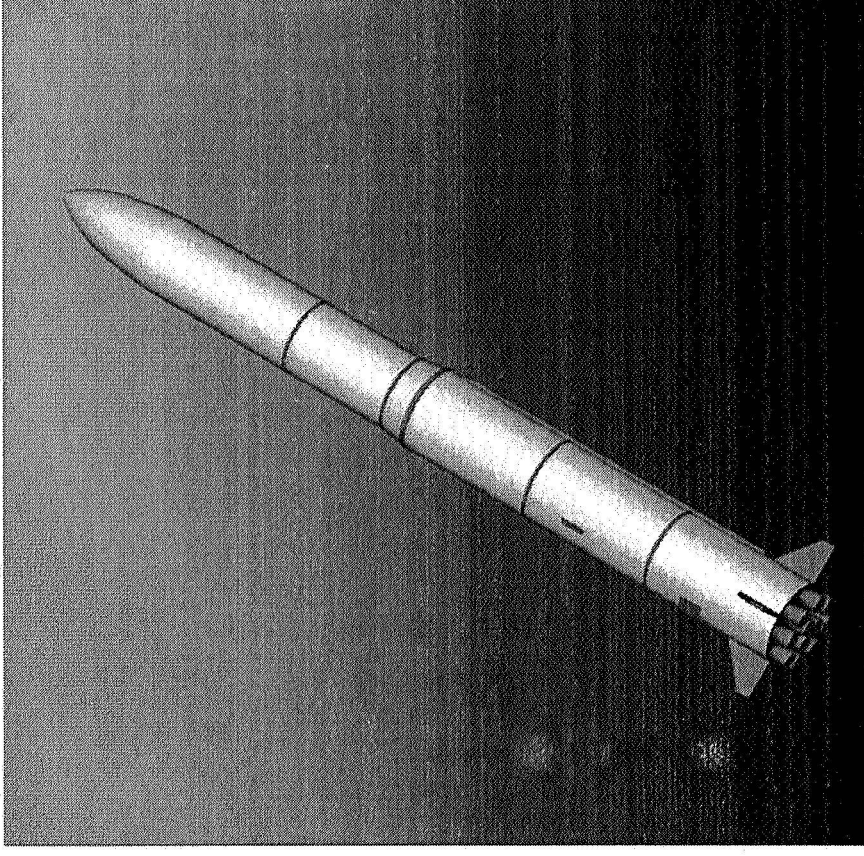
Swarm



Moonraker

Colossus

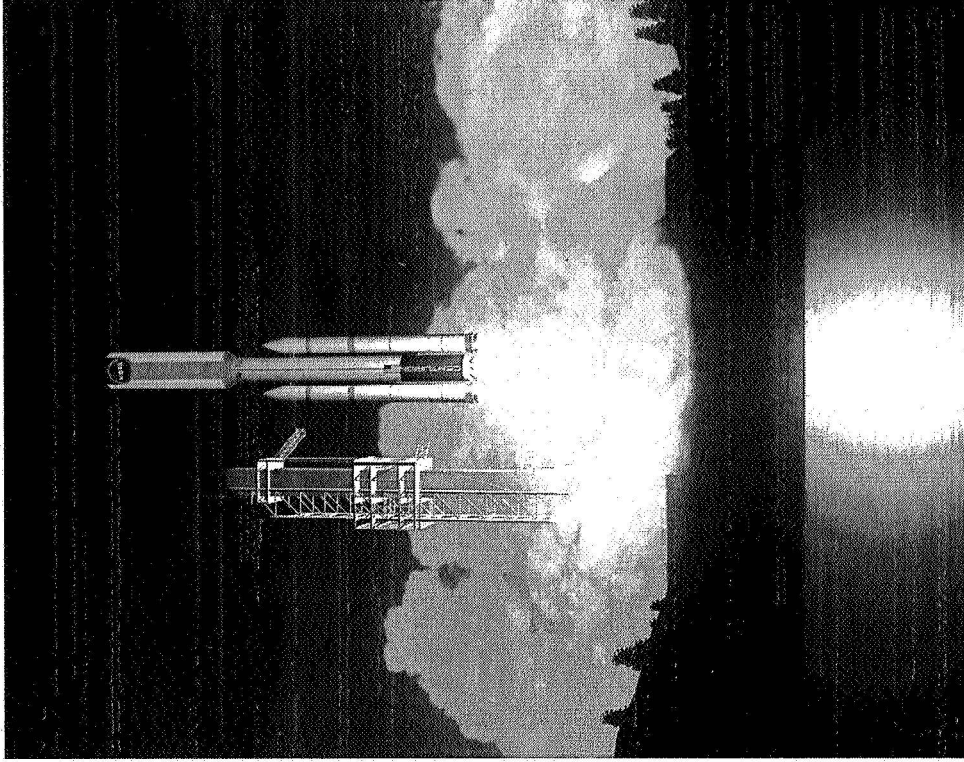
- Heavy Lift Launch Vehicle (HLLV)
- Scalable from 40-120 MT
- LOX/LH2 Propellant
- O/F Ratio = 6.5 (tech defined)
- $I_{sp_{vac}} = 454$ s (tech defined)
 - SSME baseline
- Same engines on core and upper stage (user input)
- Desired orbit (user choice)





Centurion

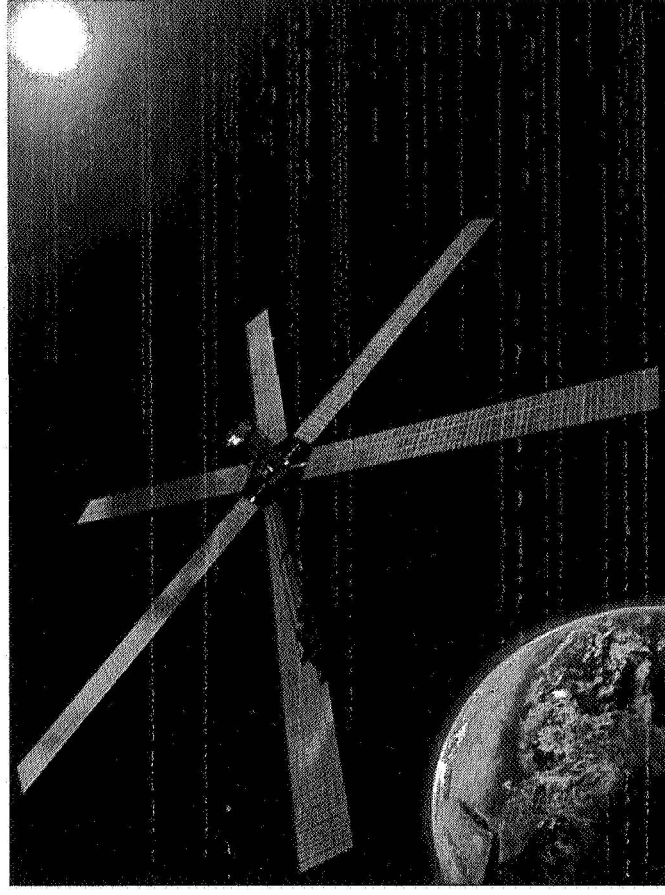
- Heavy Lift Launch Vehicle (HLLV)
- 3 payload classes
 - 35/100/142 MT
- LOX/RP first stage propellant
 - RD-180 core engines
- LOX/LH2 upper stage propellant
 - New upper stage engine
- 5 segment STS derived solid boosters
 - 0 for 35 MT class
 - 2 for 100 MT class
 - 4 for 142 MT class
- Aluminum-Lithium tanks
- Composite intertank and fairing





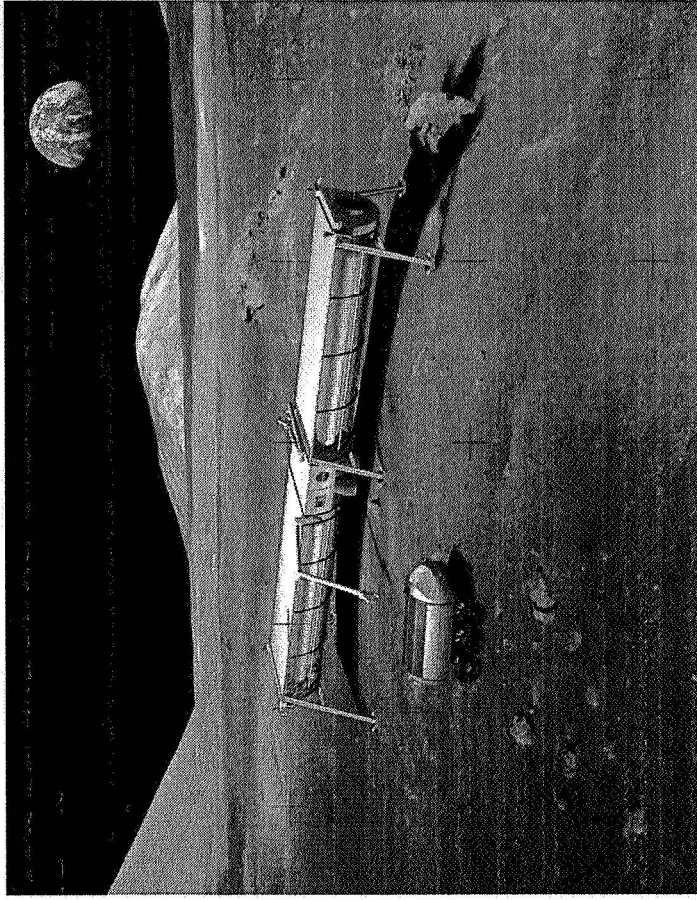
SWARM

- Solar electric powered in-space tug
- Scalable from 0-100 MT of payload from LEO to LLO
- $I_{sp_{vac}}$: 2000 - 10,000 s
- Time of Flight: 2 – 180 days
- Technologies include:
 - Varying solar panel selections
 - Different thruster types
 - Multiple propellant types



Artemis

- Lunar lander travels from LLO (100 km) to lunar surface
- Carries up to 50,000 kg payload
- LOX/LH2 Propellant
- O/F Ratio = 5.5 (tech defined)
- $I_{sp_{vac}} = 449$ s (tech defined)
 - RL-10 derived engine
- Reusability based upon user input
- Engine out capability (user input)
- Tank margin (user input)
- Refueling option (user input)



Conclusions

- ATLAS provides technology portfolio managers a tool with which to analyze the impact of technology choices on cost and performance metrics of a space exploration architecture
- ATLAS uses a combination of physics based vehicle models and historically calibrated cost models to simulate future vehicle/programmatic performance

